



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/966,475	09/28/2001	Morrie Gasser	EMC01-08(01043)	6483
	7590	12/15/2004	EXAMINER	
Barry W. Chapin, Esq. CHAPIN & HUANG, L.L.C. Westborough Office Park 1700 West Park Drive Westborough, MA 01581			PESIN, BORIS M	
			ART UNIT	PAPER NUMBER
			2174	
DATE MAILED: 12/15/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

DETAILED ACTION

Response to Amendment

This communication is responsive to Amendment A, filed 07/16/2004.

Claims 1-34 are pending in this application. Claims 1, 17, 33, and 34 are independent claims. In the Amendment A, Claims 1, 10, 17, 26, 33, and 34 were amended. This action is made Final.

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-8, 11-24, and 27-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Microsoft Windows NT (Screen Shots) in view of Griesmer (US 5923328).

In regards to claim 1, Windows NT teaches a method for representing a resource in a computing system environment, the method comprising the steps of: creating an object to represent a resource in the computing system environment (See Figure 2, Creating a shortcut); assigning an object identifier to the object, the object identifier including at least a simple name of the object and a home of the object (See Figure 4, The user gets to select the name for the object, the home is inherently selected by the system where the object was created); displaying at least one representation of the object on a graphical user interface (See Figure 5), each of the at least one representation of the object including the simple name of the object (See Figure 5). Microsoft Windows NT (Screen Shots) does not teach determining whether a home condition exists for one of the at least one representation of the object displayed on the graphical user interface; and if a home condition exists for one of the at least one representation of the object displayed on the graphical user interface, including the home of the object in the representation of the one of the one of the at least one representation of the object displayed on the graphical user interface, and if the home condition does not exist, omitting the home of the object from the representation of the one of the at least one representation of the object displayed on the graphical user interface. Griesmer teaches, determining whether a home condition exists for one of the at least one representation of the object displayed on the graphical user interface

Art Unit: 2174

(Figure 2, "Time" is a directory used several times in Element 100. Sometimes it includes a suffix, but when the directory is determined to be local, it does not have a suffix.); and if a home condition exists for one of the at least one representation of the object displayed on the graphical user interface, including the home of the object in the representation of the one of the one of the at least one representation of the object displayed on the graphical user interface, and if the home condition does not exist, omitting the home of the object from the representation of the one of the at least one representation of the object displayed on the graphical user interface (Figure 2, "Time" is a directory used several times Element in 100. Sometimes it includes a suffix, but when the directory is determined to be local, it does not have a suffix.). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Windows NT (Screen Shots) with the teachings of Griesmer and include a method to determine whether to add a suffix to an object based on whether it has a home condition or not with the motivation to provide the user a simple means of differentiating objects with the same name.

In regards to claim 2, Windows NT teaches a method wherein the step of creating an object includes the step of: associating the object with at least one location within an object hierarchy such that the object becomes a child object of at least one parent object in the object hierarchy, the object hierarchy representing relationships between resources in the computing system environment which are represented by objects in the object hierarchy (See Figure 6, Element 1); wherein the at least one location to which the object is associated in the object hierarchy includes a home

location identifying a home object in object hierarchy under which the object is initially associated as a child object (See Figure 6, "Autoexec.bat" is a file under the Desktop directory), so as to define a home context for the object; and wherein the step of assigning the object identifier assigns the home of the object to be the home location identifying the home object for that object in object hierarchy (See Figure 6, the Desktop is defined to be the home of the object).

In regards to claim 3, Windows NT teaches a method wherein the step of assigning an object identifier to the object comprises the steps of: receiving a simple name for the object to uniquely identify that object (See Figure 4). Windows NT does not teach assigning a suffix to the home of the object if the home of the object is not unique in the computing system environment, such that object identifiers for objects having a home that is not unique will be different from each other based on the suffix. Griesmer teaches assigning a suffix to the home of the object if the home of the object is not unique in the computing system environment, such that object identifiers for objects having a home that is not unique will be different from each other based on the suffix (Figure 2, "Time" is a directory used several times Element in 100. Sometimes it includes a suffix, but when the directory is determined to be local, it does not have a suffix).

In regards to claim 4, Windows NT teaches a method wherein the home location to which the object is associated is a simple name included in an object identifier assigned to the home object associated with that home location, such that if a home condition exists, the at least one representation of the object displayed on the graphical

Art Unit: 2174

user interface includes the simple name of the object followed by the simple name of the home object associated with the home location of the object (See Figure 1, Element 1, There are 2 different ports with the name HPFAX and in order to differentiate the two ports, Windows puts a prefix with the home location of the port).

In regards to claim 5, Windows NT and Griesmer teach all the limitations of claim 2. Windows NT (Screen Shots) does not teach a method wherein the step of a displaying at least one representation of the object on a graphical user interface comprises the step of: displaying the object hierarchy on the graphical user interface to convey the relationships between resources in the computing system environment, such that certain of the at least one location of the object in the object hierarchy is displayed on the graphical user interface; and wherein a home condition exists for one of the at least one representation of the object displayed on the graphical user interface if displaying that representation of the object at that location in the object hierarchy in the graphical user interface causes one of: i) the object to be displayed out of a home context of the object; and ii) the object to be displayed non-uniquely in a context in which the object is displayed; such that the occurrence of a home condition causes the one of the at least one representation of the object to be displayed in a qualified manner in that location in the object hierarchy in the graphical user interface. Griesmer teaches a method wherein the step of a displaying at least one representation of the object on a graphical user interface comprises the step of: displaying the object hierarchy on the graphical user interface to convey the relationships between resources in the computing system environment, such that certain of the at least one location of the object in the

object hierarchy is displayed on the graphical user interface (Figure 2); and wherein a home condition exists for one of the at least one representation of the object displayed on the graphical user interface if displaying that representation of the object at that location in the object hierarchy in the graphical user interface causes one of: i) the object to be displayed out of a home context of the object (Figure 2); and ii) the object to be displayed non-uniquely in a context in which the object is displayed; such that the occurrence of a home condition causes the one of the at least one representation of the object to be displayed in a qualified manner in that location in the object hierarchy in the graphical user interface (Figure 2, "Time" object).

In regards to claim 6, Windows NT and Griesmer teach all the limitations of claim 2. Windows NT (Screen Shots) does not teach a method wherein a home condition exists if a user of the graphical user interface indicates that representations of objects are to be displayed on the graphical user interface in a qualified manner, such that the at least one representation of the object displayed on the graphical user interface in a qualified manner includes the simple name of the object followed by the home of the object (See Figure 1, Element 1, the home of the object is followed by the name of the object). Griesmer teaches a method wherein a home condition exists if a user of the graphical user interface indicates that representations of objects are to be displayed on the graphical user interface in a qualified manner, such that the at least one representation of the object displayed on the graphical user interface in a qualified manner includes the simple name of the object followed by the home of the object (Figure 2, "Time" is followed by the home of the object).

In regards to claim 7, Windows NT teaches a method wherein the home object in the object hierarchy, for which the home location is identified by the home of the object, is a first non-transparent ancestral home object in the object hierarchy (See Figure 6, The Desktop is the ancestral home object).

In regards to claim 8, Windows NT teaches a method wherein the object is a group object created to represent a plurality of related resources in the computing system environment, such that objects in the object hierarchy below the group object share a common relationship to each other identified by the group object (See Figure 7, Element 1, the "New Folder" is a group object that can represent a plurality of related resources).

In regards to claim 11, Windows NT and Griesmer teach all the limitations of claim 2. Windows NT does not teach a method wherein there are a plurality of objects represented in the object hierarchy and wherein the relationships between objects represented in the object hierarchy include functional relationships and organizational relationships between certain of the objects represented in the object hierarchy (See Figure 6); and wherein the step of displaying at least one representation of the object on a graphical user interface includes the step of displaying the object hierarchy on the graphical user interface to convey the functional and organizational relationships between resources in the computing system environment (See Figure 6). Griesmer teaches a method wherein there are a plurality of objects represented in the object hierarchy and wherein the relationships between objects represented in the object hierarchy include functional relationships and organizational relationships between

certain of the objects represented in the object hierarchy (Figure 5); and wherein the step of displaying at least one representation of the object on a graphical user interface includes the step of displaying the object hierarchy on the graphical user interface to convey the functional and organizational relationships between resources in the computing system environment (Figure 5).

In regards to claim 12, Windows NT teaches a method wherein: the plurality of objects represented in the object hierarchy represent resources in the computing system environment including storage system resources, computing system resources, and storage area network resources (See Figure 6, its shows storage and computing system resources); wherein a user of the graphical user interface can manage resources associated with object in the object hierarchy via selection of representations of objects in the object hierarchy displayed on the graphical user interface (The user can manage the resources via Figure 8); and wherein all objects containing a representation in the graphical user interface have a simple name and a home that combine to define a single name space for all objects in the computing system environment irrespective of what those objects represent (See Figure 6, all objects have a name, and the home is the current directory).

In regards to claim 13, Windows NT and Griesmer teach all the limitations of claim 2. Windows NT (Screen Shots) does not teach a method wherein there are multiple representations of the same object within the object hierarchy and wherein representations of the object that appear in the graphical user interface in a non-home context are displayed in the graphical user interface in a fully qualified manner so as to

indicate the simple name of the object followed by the home of the object (See Figure 1, Element 1, except that the home is followed by the simple name). Griesmer teaches a method wherein there are multiple representations of the same object within the object hierarchy and wherein representations of the object that appear in the graphical user interface in a non-home context are displayed in the graphical user interface in a fully qualified manner so as to indicate the simple name of the object followed by the home of the object (Figure 2, "Time" object).

In regards to claim 14, Windows NT teaches a method further including the steps of moving the object to a new home location in the object hierarchy such that the object has a new home context (See Figure 8, the "Send To" command, or the user may simply cut and copy the object into a new directory (i.e. new home)); and determining if the simple name for the object uniquely identifies the object in the new home context for the object with respect to other object having the same home context, and if the simple name for the object does not uniquely identify the object in the new home context for the object, altering the simple name to provide a unique simple name for the object in the new home context (See Figure 8, the rename command).

In regards to claim 15, Windows NT teaches a method wherein the step of altering comprises appending a suffix to the end of the simple name of the object such that the simple name uniquely identifies the object in the new home context (See Figure 1, Element 1, the name is appended to the home path to create a unique name).

In regards to claim 16, Windows NT and Griesmer teach all the limitations of claim 1. Windows NT (Screen Shots) do not teach a method wherein the object can be

Art Unit: 2174

represented in a fully qualified manner to indicate a specific instance of the resource associated with that object by representing the object with the simple name of the object followed by the home of the object. Griesmer teaches a method wherein the object can be represented in a fully qualified manner to indicate a specific instance of the resource associated with that object by representing the object with the simple name of the object followed by the home of the object (Figure 2, "Time" object).

Claim 17 is in the same context as claim 1; therefore it is rejected under similar rationale. The hardware aspects of the claim are inherently in Windows NT.

Claim 18 is in the same context as claim 2; therefore it is rejected under similar rationale.

Claim 19 is in the same context as claim 3; therefore it is rejected under similar rationale.

Claim 20 is in the same context as claim 4; therefore it is rejected under similar rationale.

Claim 21 is in the same context as claim 5; therefore it is rejected under similar rationale.

Claim 22 is in the same context as claim 6; therefore it is rejected under similar rationale.

Claim 23 is in the same context as claim 7; therefore it is rejected under similar rationale.

Claim 24 is in the same context as claim 8; therefore it is rejected under similar rationale.

Claim 27 is in the same context as claim 11; therefore it is rejected under similar rationale.

Claim 28 is in the same context as claim 12; therefore it is rejected under similar rationale.

Claim 29 is in the same context as claim 13; therefore it is rejected under similar rationale.

Claim 30 is in the same context as claim 14; therefore it is rejected under similar rationale.

Claim 31 is in the same context as claim 15; therefore it is rejected under similar rationale.

Claim 32 is in the same context as claim 16; therefore it is rejected under similar rationale.

Claim 33 is in the same context as claim 1; therefore it is rejected under similar rationale.

Claim 34 is in the same context as claim 1; therefore it is rejected under similar rationale.

Allowable Subject Matter

Claims 9 and 25 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Prior art does not teach creating a transparent group object and further having its children identify its home (the transparent group's home) as their home.

Art Unit: 2174

Claims 10 and 26 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112; second paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

Prior art does not teach creating a terminal group object and further having its children identify its home (the terminal group's home) as their home.

Response to Arguments

Applicant's arguments with respect to claims 1-8, 11-24, and 27-34 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

Art Unit: 2174

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Inquiry

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Boris Pesin whose telephone number is (571) 272-4070. The examiner can normally be reached on Monday-Friday, 9AM - 6:30 PM, except every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on (571) 272-4063. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

BP

Kristine Kincaid
KRISTINE KINCAID
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100